A design document, giving the overall structure of your solution, showing the ***classes*** and ***methods*** as a diagram or table, with ***a brief (one sentence) description of the intention*** of each.

Class: backend	
Intention: backend system of the banking system	
Methods	**Intentions**
readOldMaster(String fileName)	Read the old master account file
readTrans(String fileName)	Read new transaction summary file(s)
updateMaster(String[] updateFile)	update master account file with new transaction summary file(s)
updateValid(String[] updateFile)	update the valid account list with new transaction summary file(s)
reverseSort()	Sort the master account in descending order
newValidList()	Create the new valid account list based on old master account + new transaction summary files
checkMaster(ArrayList<String> result)	Check constraints of the new master account file
writeMaster(String path)	write new master account file
writeValid(String path)	write new valid account list
main(String args[])	Main function of the class

Class: *accounts*	
Intention: The class to store account information including balance	
Methods	**Intentions**
accounts(int number, long balance, String name)	Constructor for the accounts class, set values to local attributes
setBalance(double number)	Call by other class in order to modify the balance
getAccountNumber()	Return the account number
getAccountBalance()	Return the account balance
getAccountName()	Return the account name
toString()	To String method

In particular, the backend class will take at least two inputs, where the first input is the old master account file, the rest of inputs will be transaction summary file(s). Through the prototype, if all constraints are met, the old master account file will be updated using the input transaction summary file(s) and sorted in reverse order according to account numbers. Also, a new valid list file will be created using the input transaction summary file(s). The program will then write the new master account file and new valid account list.